The Mathematica® Journal

Stochastic Simulation and Parameter Estimation of the FitzHugh–Nagumo Model

Béla Paláncz

This article illustrates how Mathematica can be employed to model stochastic processes via stochastic differential equations to compute trajectories and their statistical features. In addition, we discuss parameter estimation of the model via the maximum likelihood method with global optimization. We consider handling modeling error, system noise and measurement error, compare the stochastic and deterministic models and use the likelihood-ratio test to verify the possible improvement provided by stochastic modeling. The Mathematica code is simple and can be easily adapted to similar problems.

Introduction

Stochastic differential equations (SDEs) have received great attention in a large number of fields, including finance, physics, system biology, biochemical processes and pharmacokinetics. SDEs serve as a natural way of introducing uncertainty into a deterministic model represented by ordinary differential equations (ODEs). In contrast to the classical approach where uncertainty only exists in the measurements, SDEs can provide a more flexible framework to account for deviation in states and parameters that describe the underlying system. For an introduction to the theory and numerical solution of SDEs, see [1].

In this article, we illustrate how Mathematica deals with the problem of simulation and parameter estimation of such systems, represented here by the FitzHugh–Nagumo model describing excitable media. The first section sets up the model equations. The second section deals with the simulation of the stochastic system and computes the trajectories, their distribution at a specified time point and the change of the statistical data of these distributions in time, namely the trajectory of the mean value and standard deviation. The third section describes the parameter estimation of the model via the maximum likelihood method (ML). The last section analyzes the goodness of fit between deterministic and stochastic models using the likelihood ratio test (LRT).

■ Modeling

The FitzHugh–Nagumo model for excitable media is a nonlinear model describing the reciprocal dependencies of the voltage $V(t)$ across an exon membrane and a recovery variable $R(t)$ summarizing outward currents. The model was developed in [2]. The model is general and can also model excitable media, for example, heart tissue. The deterministic ODE model (white-box model) is described by the following system of ordinary differential equations:

$$\frac{dV(t)}{dt} = \gamma \left(V(t) - \frac{V(t)^3}{3} + R(t) \right), \tag{1}$$

$$\frac{dR(t)}{dt} = -\frac{1}{\gamma} (V(t) - \alpha + \beta\, R(t)), \tag{2}$$

with parameters α, β and γ and initial condition $V(0) = -1$ and $R(0) = 1$.

White-box models are mainly constructed on the basis of knowledge of physics about the system. Solutions to ODEs are deterministic functions of time, and hence these models are built on the assumption that the future value of the state variables can be predicted exactly. An essential part of model validation is the analysis of the residual errors (the deviation between the true observations and the one-step predictions provided by the model). This validation method is based on the fact that a correct model leads to uncorrelated residuals. This is rarely obtainable for white-box models. Hence, in these situations, it is not possible to validate ODE models using standard statistical tools. However, by using a slightly more advanced type of equation, this problem can be solved by replacing ODEs with SDEs that incorporate the stochastic behavior of the system: modeling error, unknown disturbances, system noise and so on.

The stochastic SDE gray-box model can be considered as an extension of the ODE model by introducing system noise:

$$dV(t) = \gamma \left(V(t) - \frac{V(t)^3}{3} + R(t) \right) dt + \sigma\, dW(t), \tag{3}$$

$$dR(t) = -\frac{1}{\gamma} (V(t) - \alpha + \beta\, R(t))\, dt + \sigma\, dW(t), \tag{4}$$

where $W(t)$ is a Wiener process (also known as Brownian motion), a continuous-time random walk. The next section carries out the numerical simulation of the SDE model using the parameter settings $\alpha = \beta = 0.2$, $\gamma = 3$ and $\sigma = 0.1$.

Simulation

Equations (3) and (4) represent an Ito-stochastic process that can be simulated in Mathematica employing a stochastic Runge–Kutta method.

```
procVR = ItoProcess[
   {𝕕V[t] == γ (V[t] - V[t]^3/3 + R[t]) 𝕕t + σ 𝕕Vw[t],
    𝕕R[t] == -1/γ (V[t] - α + β R[t]) 𝕕t + σ 𝕕wR[t]},
   {V[t], R[t]}, {{V, R}, {-1, 1}}, t,
   {wV ≈ WienerProcess[], wR ≈ WienerProcess[]}
  ];
```

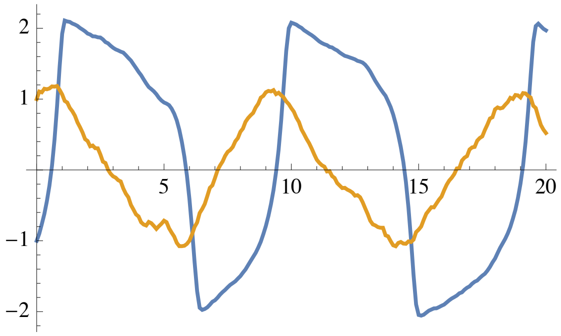
```
param = {α → 0.2, β → 0.2, γ → 3, σ → 0.1};
```

Single Realization

First, a single realization is simulated in the time interval $0 \leq t \leq 20$.

```
dataVR = RandomFunction[procVR /. param, {0, 20, 0.1},
   Method → "StochasticRungeKutta"];
```

```
ListLinePlot[dataVR]
```



▲ **Figure 1.** The trajectories of the state variables $V(t)$ (blue) and $R(t)$ (brown) in the case of a single realization of the Ito process.

The values of V and R can be found at any time point. Here is an example.

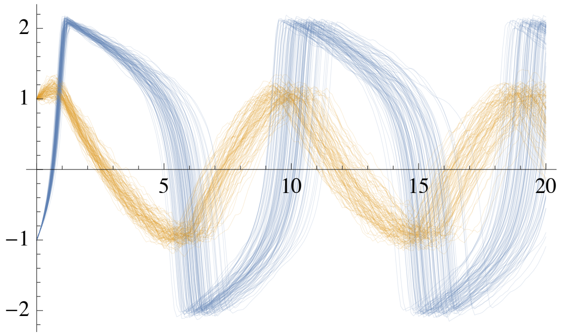
```
dataVR["SliceData", τ] /. τ → 12.
```

```
{{1.63315, -0.25604}}
```

□ Many Realizations

This computes the trajectories for 100 realizations.

```
Show[Map[
   ListLinePlot[#,
      PlotStyle → Directive[Opacity[0.2], Thin]] &,
   Table[RandomFunction[procVR /. param, {0., 20., 0.1}, 1],
     {100}]
  ]]
```



▲ **Figure 2.** The trajectories of the state variables $V(t)$ (blue) and $R(t)$ (brown) in the case of 100 realizations of the Ito process.

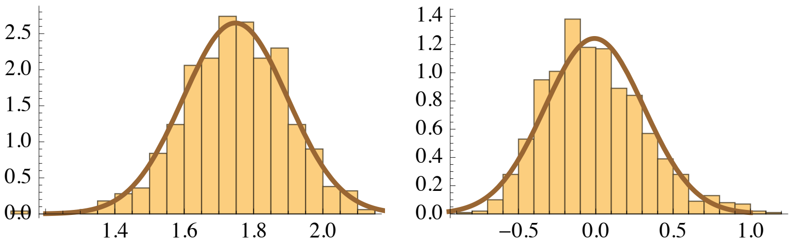
Slice distributions of the state variables can be computed at any time point. First, let us simulate the trajectories in a slightly different and faster way, now using 1000 realizations.

```
td = RandomFunction[procVR /. param, {0., 20., 0.1}, 1000]
```

```
TemporalData[ ⊞  Time: 0. to 20.
                  Data points: 201 000       Paths: 1000 ]
```

At the time $\tau = 12$, we can compute the mean and standard deviation for both state variables $V(\tau)$ and $R(\tau)$, as well as their histograms.

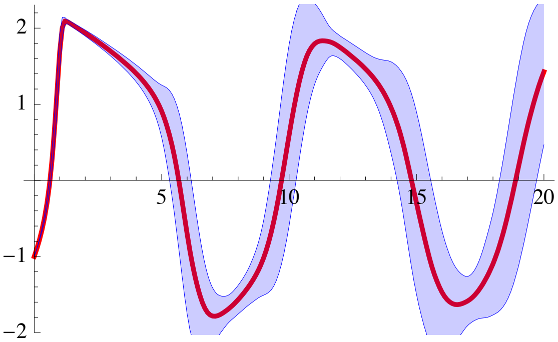
```
Module[
  {τ = 12, μV, μR, σV, σR, dataτV, dataτR},
  {μV, μR} = Mean[td[τ]];
  {σV, σR} = StandardDeviation[td[τ]];
  {dataτV, dataτR} =
   {Map[#[[1]] &, td[τ][[2]][[2]]],
    Map[#[[2]] &, td[τ][[2]][[2]]]};
  GraphicsGrid[{{
     Show[{
       Histogram[dataτV, Automatic, "PDF"],
       Plot[PDF[NormalDistribution[μV, σV], t],
       {t, 1.2, 2.5}, PlotStyle → {Thick, Brown}]
     }],
    Show[{
      Histogram[dataτR, Automatic, "PDF"],
      Plot[PDF[NormalDistribution[μR, σR], t],
       {t, -0.95, 1.}, PlotStyle → {Thick, Brown}]
     }]
   }}, ImageSize → 500]
]
```



▲ **Figure 3.** The distribution of $V(\tau)$ (left) and $R(\tau)$ (right) in the case of 100 realizations of the Ito process, at time $\tau = 12$.

The mean value and the standard deviation of the trajectories along the simulation time $0 \leq t \leq 20$ can also be computed and visualized. (This may take a few minutes.)

```
Show[{
  Plot[Mean[td[t]][[1]], {t, 0, 20},
   PlotStyle → {Red, Thick}],
  Plot[{Mean[td[t]][[1]] + StandardDeviation[td[t]][[1]],
    Mean[td[t]][[1]] - StandardDeviation[td[t]][[1]]},
   {t, 0, 20}, Filling → {1 → {2}},
   PlotStyle → {{Blue, Thin}, {Blue, Thin}}, PlotRange → All]
 }]
```



▲ **Figure 4.** $\mu_V(t)$, the mean value of $V(t)$ (red) with its standard deviation, $\mu_V(t) \pm \sigma_V(t)$ (blue).

Parameter Estimation

Parameter estimation is critical since it determines how well the model compares to the measurement data. The measurement process itself may also have serially uncorrelated errors due to the imperfect accuracy and precision of the measurement equipment.

Measurement Values

Write the measurement equation as

$$y_k = V(t_k) + e_k, \tag{5}$$

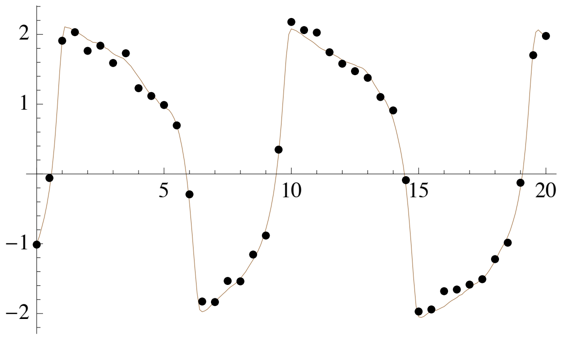
where $e_k \sim \mathcal{N}(0, \sigma_m)$. The voltage $V(t)$ is assumed to be sampled between $t = 0$ and $t = 20$ at discrete time points t_k, where $k = 0, 1, 2, \ldots, N$, $N = 40$, with an additive measurement noise σ_m. To get $V(t_k)$, we consider a single realization and sample it at time points t_k. Then we add white noise with $\sigma_m = 0.1$.

```
SV[t_] := First[Flatten[dataVR["SliceData", t] /. param]]

pSV = Plot[SV[t], {t, 0, 20}, PlotStyle → {Thin, Brown}];
dataSV = Module[
   {σm = 0.1, ek},
   ek = RandomVariate[NormalDistribution[0, σm], 41];
   Table[{i/2, SV[i/2] + ek[[i + 1]]}, {i, 0, 40}]
  ];

pVm = ListPlot[dataSV, PlotStyle → {Black, PointSize[0.015]}];

Show[{pSV, pVm}, PlotRange → All]
```



▲ **Figure 5.** A single realization of $V(t)$ (brown) and the simulated measured values y_k (black points).

□ Likelihood Function

As we have seen, the solutions to SDEs are stochastic processes that are described by probability distributions. This property allows for maximum likelihood estimation. Let the deviation of the measurements from the model be

$$\epsilon_k(\theta) = y_k - \mu_V(t_k, \theta), \tag{6}$$

where $\theta = \{\alpha, \beta, \gamma\}$. Assuming that the density function of ϵ_k can be approximated reasonably well by Gaussian density, the likelihood function to be maximized is

$$\mathcal{L}(\theta) = \frac{1}{2\pi}\exp\left(-\sum_{k=1}^{N}\epsilon_k\,(\theta)^2\right). \tag{7}$$

For computation, we use its logarithm. Now we accept the model parameter γ and estimate α and β from the SDE model, employing the maximum likelihood method.

```
parm = {γ → 3, σ → 0.1};
```

Here are the values of the y_k.

```
yk = Transpose[dataSV][[2]]
```

```
{-1.01253, -0.058196, 1.90877, 2.03138, 1.76385,
 1.83778, 1.58995, 1.7294, 1.22855, 1.11723,
 0.988036, 0.695723, -0.292415, -1.82789, -1.83584,
 -1.53323, -1.53899, -1.15499, -0.882498, 0.348372,
 2.1779, 2.06111, 2.0256, 1.74369, 1.57974, 1.47242,
 1.37736, 1.10221, 0.910284, -0.0886507, -1.97125,
 -1.94147, -1.68002, -1.65612, -1.58551, -1.50796,
 -1.22078, -0.984904, -0.125907, 1.70185, 1.97732}
```

This defines the logarithmic likelihood function.

```
Log𝓛[{α_?NumericQ, β_?NumericQ}] := Module[{proc, tδ, ℰ},
  proc = ItoProcess[
    {𝕕V[t] == γ (V[t] - V[t]^3/3 + R[t]) 𝕕t + σ 𝕕wV[t],
     𝕕R[t] == -1/γ (V[t] - α + β R[t]) 𝕕t + σ 𝕕wR[t]},
    {V[t], R[t]}, {{V, R}, {-1, 1}}, t,
    {wV ≈ WienerProcess[], wR ≈ WienerProcess[]}];
  tδ = RandomFunction[proc /. parm, {0., 20., 0.1}, 100];
  ℰ = yk - Table[Mean[tδ[t]][[1]], {t, 0, 20, 1/2}];
  LogLikelihood[NormalDistribution[0, 1], ℰ]]
```

□ Maximization of the Likelihood Function

The optimization of the likelihood function is not an easy task, since the objective function is often flat, with nondifferentiable terms and many local extrema. In addition, the model takes a long time to evaluate. Instead of using direct global optimization, first we compute the values of the objective function on a 25×25 grid to use parallel computation to speed up the evaluation. We are looking for the parameters in the range $-0.1 \leq \alpha, \beta \leq 0.5$. Let us create the grid points.

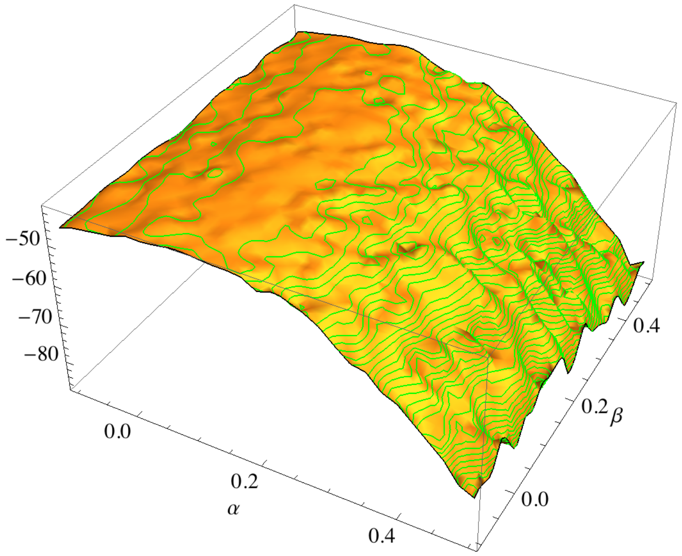
```
XY = Table[{-0.1 + i 0.025, -0.1 + j 0.025}, {i, 0, 24},
   {j, 0, 24}];
```

This computes the function values at the grid points in parallel.

```
DistributeDefinitions[Logℒ, yk, parm, XY];
LaunchKernels[8];
G = ParallelTable[Logℒ[XY[[i, j]]], {i, 25}, {j, 25}];
CloseKernels[];
```

Now apply interpolation for the grid point data and visualize the likelihood function.

```
XYZ = MapThread[{#1, #2} &, {Flatten[XY, 1], Flatten[G]}];
F = Interpolation[XYZ, InterpolationOrder → 3];
Plot3D[F[α, β], {α, -0.1, 0.5}, {β, -0.1, 0.5},
 AxesLabel → Automatic,
 MeshFunctions → Function @@@ {{{α, β, z}, F[α, β]}},
 Mesh → 40, MeshStyle → {Orange, Green},
 BoxRatios → {1, 1, 0.5}]
```



▲ **Figure 6.** The likelihood function of the parameter estimation problem.

Employing different global optimization methods, we compute the parameters. The third one generates a warning message that we suppress with `Quiet`.

```
Module[{uvR, uvS, uvN},
 uvD = NMaximize[{F[u, v], -0.1 < u < 0.5, -0.1 < v < 0.5},
   {u, v}, Method -> "DifferentialEvolution"];
 uvR = NMaximize[{F[u, v], -0.1 < u < 0.5, -0.1 < v < 0.5},
   {u, v}, Method → {"RandomSearch", "SearchPoints" → 1000}];
 uvS = Quiet@NMaximize[{F[u, v], -0.1 < u < 0.5, -0.1 < v < 0.5},
    {u, v}, Method -> "SimulatedAnnealing"];
 uvN = NMaximize[{F[u, v], -0.1 < u < 0.5, -0.1 < v < 0.5},
   {u, v}, Method -> "NelderMead"];
 Text@
  TableForm[
   {{"DifferentialEvolution", "RandomSearch",
     "SimulatedAnnealing", "NelderMead"}, {},
    Flatten /@ {uvD, uvR, uvS, uvN}}]
]
```

```
DifferentialEvolution   RandomSearch    SimulatedAnnealing   NelderMead

–41.1421                –41.0766        –41.2587             –41.3695
u → 0.141924            u → 0.175       u → 0.121824         u → 0.125
v → 0.270284            v → 0.0640822   v → 0.05             v → –0.0426
```

There are many local maxima. Let us choose the global one.

Simulation with the Estimated Parameters

Here is a new parameter list.

```
paramN = {α → u, β → v, γ → 3, σ → 0.1} /. uvD[[2]]
```

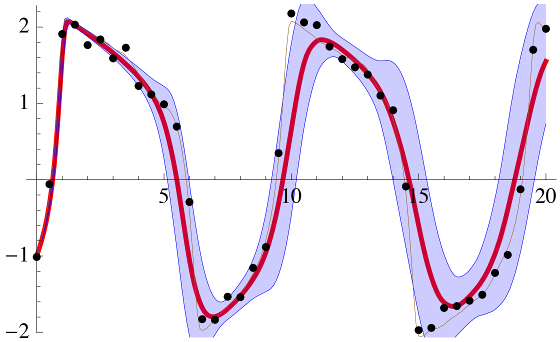
```
{α → 0.141924, β → 0.270284, γ → 3, σ → 0.1}
```

This computes 500 realizations.

```
τd = RandomFunction[procVR /. paramN, {0., 20., 0.1}, 500];
```

This visualizes the result and the measurement data.

```
Show[{
  Plot[Mean[τd[t]][[1]], {t, 0, 20},
   PlotStyle → {Red, Thick}],
  Plot[{Mean[τd[t]][[1]] + StandardDeviation[τd[t]][[1]],
    Mean[τd[t]][[1]] - StandardDeviation[τd[t]][[1]]},
   {t, 0, 20}, Filling → {1 → {2}},
   PlotStyle → {{Blue, Thin}, {Blue, Thin}},
   PlotRange → All],
  pSV, pVm
 }]
```



▲ **Figure 7.** The simulated $V(t)$ process with the estimated parameters as in Figure 4.

Deterministic versus Stochastic Modeling

In the previous section, Parameter Estimation, we illustrated the technique of stochastic modeling. The measurement data was simulated with the correct model parameters ($\gamma = 3$), no assumed modeling error, existing system noise $\sigma = 0.1$ and measurement error $\sigma_m = 0.1$. The model to be fitted had two free parameters, α and β, since the system noise $\sigma = 0.1$ was chosen for the model.

Now consider a different situation. Suppose that we have modeling error, since the measured values are simulated with $\gamma = 2.5$; however, in the model to be fitted, we use $\gamma = 3$. In addition, let us increase the measurement error to $\sigma_m = 0.2$. To compare the efficiencies of the deterministic and stochastic models, we should handle σ as a free parameter in the stochastic model to be fitted. So then we have three free parameters to be estimated: α, β and σ. Let us carry out the parameter estimation for different values of σ.

σ	$-\,log\mathcal{L}\,(\alpha,\beta)$	α	β
0.00	61.060	–0.037	0.230
0.10	54.938	–0.004	0.300
0.25	52.761	0.151	0.125
0.40	54.687	0.044	0.150

▲ **Table 1.** The results of the parameter estimation for different system noises.

The results can be seen in Table 1; $\sigma = 0$ corresponds to the deterministic model. In order to demonstrate that the stochastic model can provide significant improvement compared with the deterministic model, the likelihood-ratio test can be applied [3]. The likelihood-ratio test has been used to compare two nested models. In our case, one of the models is the ODE model, with fewer parameters than the SDE model. The null hypothesis is that the two models are basically the same. The test statistic is

$$\mathcal{R} = 2\,((-log\mathcal{L}\,(\alpha,\beta))_{\mathcal{D}} - (-log\mathcal{L}\,(\alpha,\beta))_{\mathcal{S}}), \tag{8}$$

where the subscripts $\mathcal{D}$ and $\mathcal{S}$ stand for the deterministic and stochastic models, respectively.

The distribution of $\mathcal{R}$ is $\chi^2(f)$, where f is the difference in the number of parameters between the two models; in our case, $f = 3 - 2 = 1$. Here is the critical value for $\chi^2(1)$ at a confidence level of 95%.

```
InverseCDF[ChiSquareDistribution[1], 0.95]
```

```
3.84146
```

If the value of $\mathcal{R}$ is less than the critical value, then the null hypothesis can be accepted. In our case, $\mathcal{R} = 2\,(61.060 - 52.761) = 16.598$. Therefore, we reject the null hypothesis, which means the SDE model can be considered as a different model providing a significant improvement compared with the ODE model.

For changing parameters to simulate the *measured values,* you should modify the content of the list `param` in the Simulation section and the value of σ_m in the Measurement Values subsection. To change the parameter of the model to be fitted, change the value of σ in the list `parm` in the Likelihood Function subsection.

In reality, the three parameters (α, β, σ) should be handled *simultaneously* during the optimization process.

Assuming there is no modeling error, using the SDE model, we can separate the measurement error from the system noise represented by the estimated σ of the fitted model; see [4].

Conclusion

The advantage of using stochastic modeling is that the actual states in the model are predicted from data. This allows the prediction to stay close to the data even when the parameters in the model are imprecise. It has been demonstrated that Mathematica can significantly support a user carrying out stochastic modeling. There are many built-in functions that help in the computation of the trajectories of stochastic differential equations and their statistics, global optimization for parameter estimation and likelihood-ratio test for model verification.

Acknowledgments

The author is grateful for the EU FP7 International Research Staff Exchange Scheme, which provided the financial support that partly facilitated his stay at the Department of Mechanical Engineering of the University of Canterbury, NZ, where this article was completed.

References

[1] E. Allen, *Modeling with Itô Stochastic Differential Equations,* Dordrecht, Netherlands: Springer, 2007.

[2] R. FitzHugh, “Impulses and Physiological States in Theoretical Models of Nerve Membrane,” *Biophysics Journal*, **1**(6), 1961 pp. 445–466. doi:10.1016/S0006-3495(61)86902-6.

[3] H. Madsen and P. Thyregod, *Introduction to General and Generalized Linear Models*, Boca Raton: CRC Press, 2010.

[4] C. W. Tornøe, J. L. Jacobsen, O. Pedersen, T. Hansen and H. Madsen, “Grey-Box Modelling of Pharmacokinetic/Pharmacodynamic Systems,” *Journal of Pharmacokinetics and Pharmacodynamics,* **31**(5), 2004 pp. 401–417. doi.10.1007/s10928-004-8323-8.

B. Paláncz, “Stochastic Simulation and Parameter Estimation of the FitzHugh–Nagumo Model,”
The Mathematica Journal, 2016. dx.doi.org/doi:10.3888/tmj.18-6.

About the Author

Dr. Béla Paláncz is professor emeritus at the Department of Photogrammetry and Geoinformatics at the Faculty of Civil Engineering of the Budapest University of Economics and Technology, in Budapest, Hungary. His main interests are mathematical modeling and numeric-symbolic computations. He is the coauthor of several books and more than a hundred papers.

Dr. Béla Paláncz:
Budapest University of Economics and Technology
1111 Műegytem rkp. 3
Budapest, Hungary
palancz@epito.bme.hu